UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

TRABAJO FIN DE GRADO

A Blog for a Start-Up

David Castander Díaz

Julio 2013

Type of document

Private

Abstract

David Castander Díaz, Grado en Ingeniería Informática, Universidad Autónoma de Madrid.

Abstract of Bachelor's Thesis Project, Submitted July 2014:

Development of Optimov ApS Technologies to create a new blog.

The aim of this thesis is to support Optimov ApS web application and to show the complete process of developing a blog where there will be some news about the Company and its field, physiotherapy.

Optimov Aps Technologies has been successfully used to cure pain suffered in many patients and to prevent new pain with exercises to increase strength and flexibility. For this reason the author decided to develop a blog site to further the growth of the company, with the purpose of make the company more popular and the application more useful.

This thesis first explains what Optimov ApS does for the patients and for the physiotherapists with a short background of the beginning process. The second stage includes a short history of blogging to explain why it is very important to have your own blog nowadays. Finally the blog is designed and developed to suit the requirements of a blog for a start-up and its coming future.

The results of the development of this blog show that Optimov ApS is looking forward to using it as soon as they are ready for its launch. The author recommends further modification to the technology before any other uses are considered.

Key words

Blog, Start-Up, Company, Physiotherapy, exercise, pain, Technologies

Tipo of document

Privado

Resumen

David Castander Díaz, Grado en Ingeniería Informática, Universidad Autónoma de Madrid.

Resumen del proyecto de Fin de Grado, entregado a July 2014:

Desde el desarrollo de tecnologías de Optimov ApS a crear un blog.

El objetivo de este Trabajo de Fin de Grado es por un lado dar apoyo al desarrollo de la aplicación web de Optimov ApS y por otro, de forma individual, hacer completamente de cero un blog para dicha empresa, donde habrá diferentes noticias sobre el campo de la fisioterapia.

Las tecnologías de Optimov ApS han sido gratamente usadas para curar diferentes dolores sufridos por muchos pacientes y también como prevención de nuevos Dolores, mediante ejercicios de fuerza y estiramientos. Por esta razón el autor decidió desarrollar un blog en vista a future crecimiento de la compañia, con el único propósito de hacer la compañia más popular y su aplicación más útil.

Este TFG, primero explica qué hace exactamente Optimov ApS para los pacientes y para los fisioterapeutas de los mismos, un pequeño fondo del inicio del blog. Posteriormente, el autor cuenta la historia sobre los blogs a modo de resumen y que sirve para entender porqué es tan importante tener un blog actualmente. Para terminar, el blog ha sido diseñado e implementado de cara a una Start-Up y su futuro más próximo.

Los resultados del desarrollado e implementación de dicho blog muestran que Optimov ApS esta deseando usarlo tan pronto como estén listos para su lanzamiento. El autor recomienda, además, la modificación de la tecnología antes de considerar cualquier otro uso.

Palabras clave

Blog, Start-Up, empresa, fisioterapia, ejercicio, dolor, tecnologías.

Agradecimientos

Thank you very much to my tutor in Copenhagen, Johannes Scheibe, for giving me the opportunity to learn by myself, doing a project for Optimov ApS. Also, many thanks to José Cerdan, the founder of Optimov ApS, for great support during the time working there.

Gracias a mi ponente o co-tutor, Simone Santini, por toda la ayuda servida y todo el apoyo incondicional mostrado.

I must thank Teresa as well; Her advice and corrections to write in formal English have been very helpful.

Gracias a mis compañeros de la carrera. Sin ellos se habría hecho mucho más duro. Ellos saben perfectamente quiénes son, ahora son amigos.

Gracias a Irene, coordinadora de ORI-EPS, por hacer posible que haya podido ser Erasmus por partida doble, y además, en dos países nórdicos.

Gracias a todos los profesores, absolutamente todos me han enseñado algo fuera y dentro de sus asignaturas. Su exigencia máxima en todo momento me ha hecho más fuerte y preparado para salir al mercado laboral.

Gracias a toda mi familia, en especial, a mis padres. Se han encargado de que no abandone en ningún momento. Seguramente sin ellos no habría podido ni siquiera empezar.

Table of Contents

1. Introduction

Optimov ApS is a Danish Company that provides on-line physiotherapy services to a variety of patients. The company was founded in March, 2012 and is currently undergoing a rapid expansion of its web presence. Based in Copenhagen, the company has a world-wide clientele, providing services to patients in tens countries.

The company provides specific services at specific times and, consequently, the usage of its web site has been characterized by a considerable discontinuity: patients come to the web site when they need a service and, by and large, ignore it when they don't. This generates retention problems as lacking a reason to keep in touch with the web site; patients have many opportunities to explore competing services. In order to increase the "permanence" of the patients, and transform the Optimov web site into a more frequently visited web site, the company is studying various add-ons that, although not generating revenues *per se*, will create a more appealing user experience, thereby increasing the effectiveness of the revenue-generating part of the site and making their services more interesting.

2. Optimov's Application

In this chapter, we describe the application in which the blog will be inserted. The part of the Optimov system that we consider here is external to this work and will be touched upon only minimally in order to describe the integration of the blog into the main web site. The application constitutes the environment in which in the blog has been developed and will function, and its structure generated some of the requirements that we had to comply with.

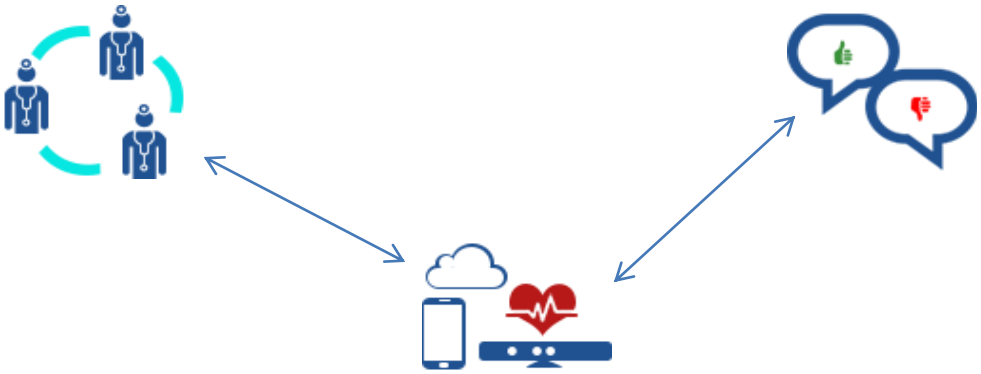
Optimov is a web tool that allows people with pain problems that require physiotherapeutic intervention to get in touch with physiotherapist, have a personalized treatment designed, and have their progress monitored on-line by the specialist.

The process begins when a person (the patient) looks for a physiotherapist because he has some pain or he wants to prevent pain. The patient sends a request for a video-appointment or an instant chat with a physiotherapist. The system gives the patient the possibility of choosing a therapist who speaks the same language, lives nearby, or whom the user trusts based on a previous interaction. At the moment, the guests of Optimov are not available to use the special services that Optimov offers. However, the patients would not have to pay anything to Optimov directly, the patients will pay for physiotherapist' services and the physiotherapist will pay to Optimov a fee to have permission to use the application for his own and for his patients.

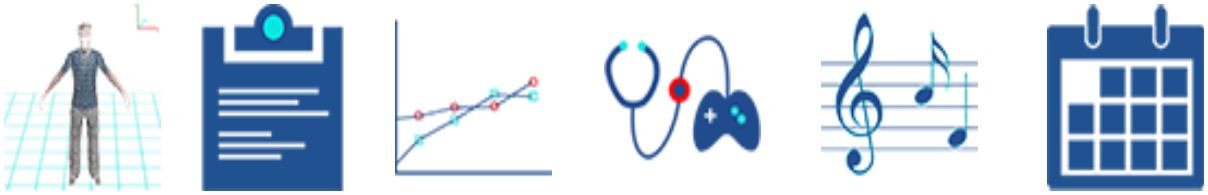
During this appointment, the system gives the therapist the possibility to ask the patient some questions to know the level of pain he is suffering or the exact moments when the user suffers that pain and when during the day the user could perform exercises to relieve it.

Subsequently, the physiotherapist sends his patient a set of exercises he designed or which were designed by Optimov's staff. The user will get a Personal 3D trainer to explain the exercises and improve his execution. Every day, the user answers some questions to rate his pain and the effectiveness of the exercises, making it easy for the physiotherapist to change them or continue with them as necessary.

The figure shows schematically the interaction of the application with the users. The patients use the application and they give some information about how they feel replying some questions. Then, the specialist can see the progress and send other exercises to the application which will send an advice to the patient. In a few words, we can say that patients and specialists use this application to interact each other.



In addition to this main service, the Optimov application offers auxiliary services such a trainer, a journal, progress graphics, exercise games, music and a calendar.



a) Specification

1) Why Optimov ApS 's business rationale

The company identified three major social and habit changes that create a potential demand for a service like that which Optimov provides:

- Work is increasingly becoming deskwork, making people move less. This results in neck and back pain as well as the degeneration of muscles.
- High workloads are making many people stressed. This can lead to burn-out and depression, if nothing is done.
- Demographic changes are overburdening the health system causing long waiting times and preventing some people from getting the treatment they need.

Optimov saw the internet a tool for mitigating some of the problems created by this situation. It created a web tool that allows patients and physiotherapist to get in touch and work together on a physiotherapy. Specifically, the Optimov tool offers the following services.

i) For Physiotherapists

- Create custom exercise sets: there is a large database of over 500 exercises, as well as pre-made templates of rehab-programs. The physioterapist can use a filter to specify body parts or training type, and add some evaluation questions that the patient has to answer before and after the exercises.

- Patient following: the physiotherapist can do video consultations with his patients and find the best exercise regime. The platform has an internal messaging system which helps to stay in touch each other.

- Share the patient journal: every physiotherapist can write a status review of the patient by taking notes during the video appointments. Then, the patients can participate in their rehabilitation processes. The notes can be used to form a training history of each patient, ensuring and uninterrupted improvement process.

- Track progress and adjust: the physioterapist is given the possibility to follow the progress of his patient, through questions and answers. In this way, the therapist can decide whether it is necessary adjust his training remotely. In the near future , it will be possible to obtain statistics from optical motion tracking and biometric sensors.

- Schedule appointments: there is the possibility to see a calendar for scheduling all appointments. Then the physiotherapist can share an exercise plan with the patient and an automated message via e-mail will be sent it to him to remain everything.

- Collaborate and share: it is very easy to share your exercises with your colleagues, creating a network of knowledge and best practices. This will give more quality to the physiotherapists' exercises, so the patients will get the best treatment possible.

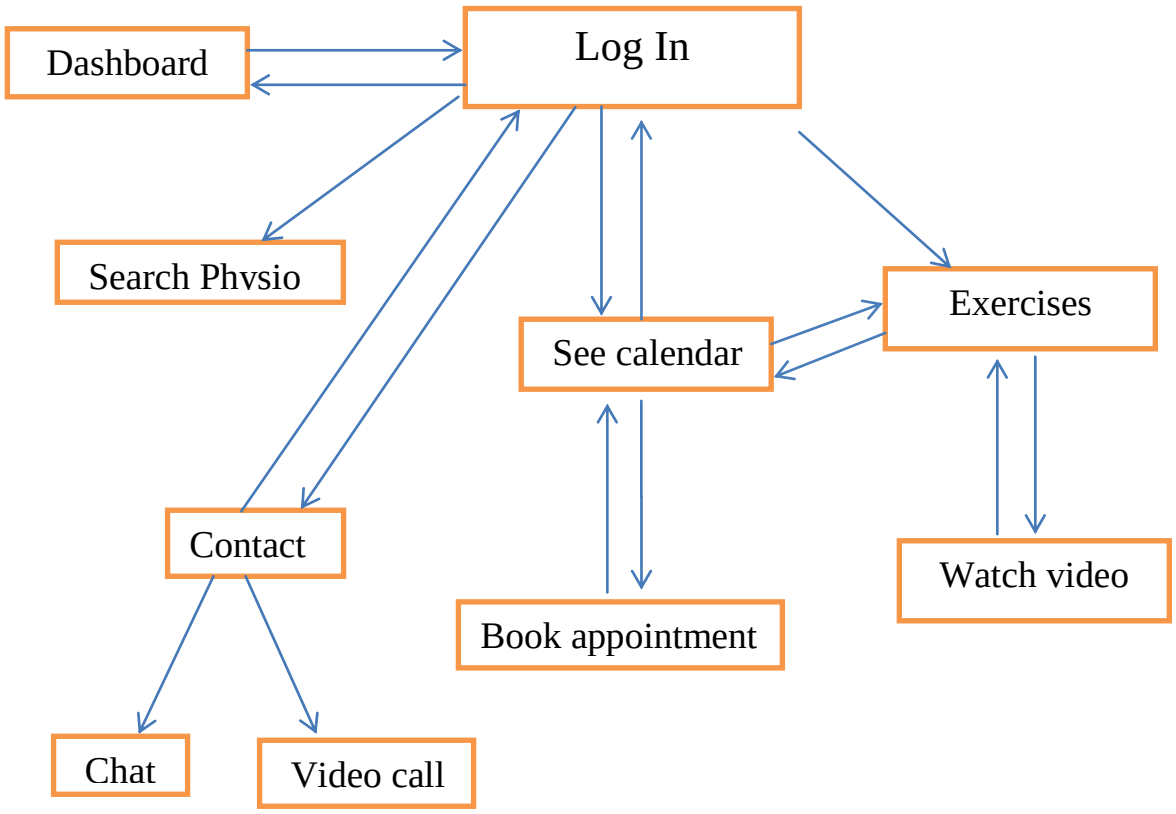
ii) Benefits for Patients

From the point of view of the patient the system offers the following functionality

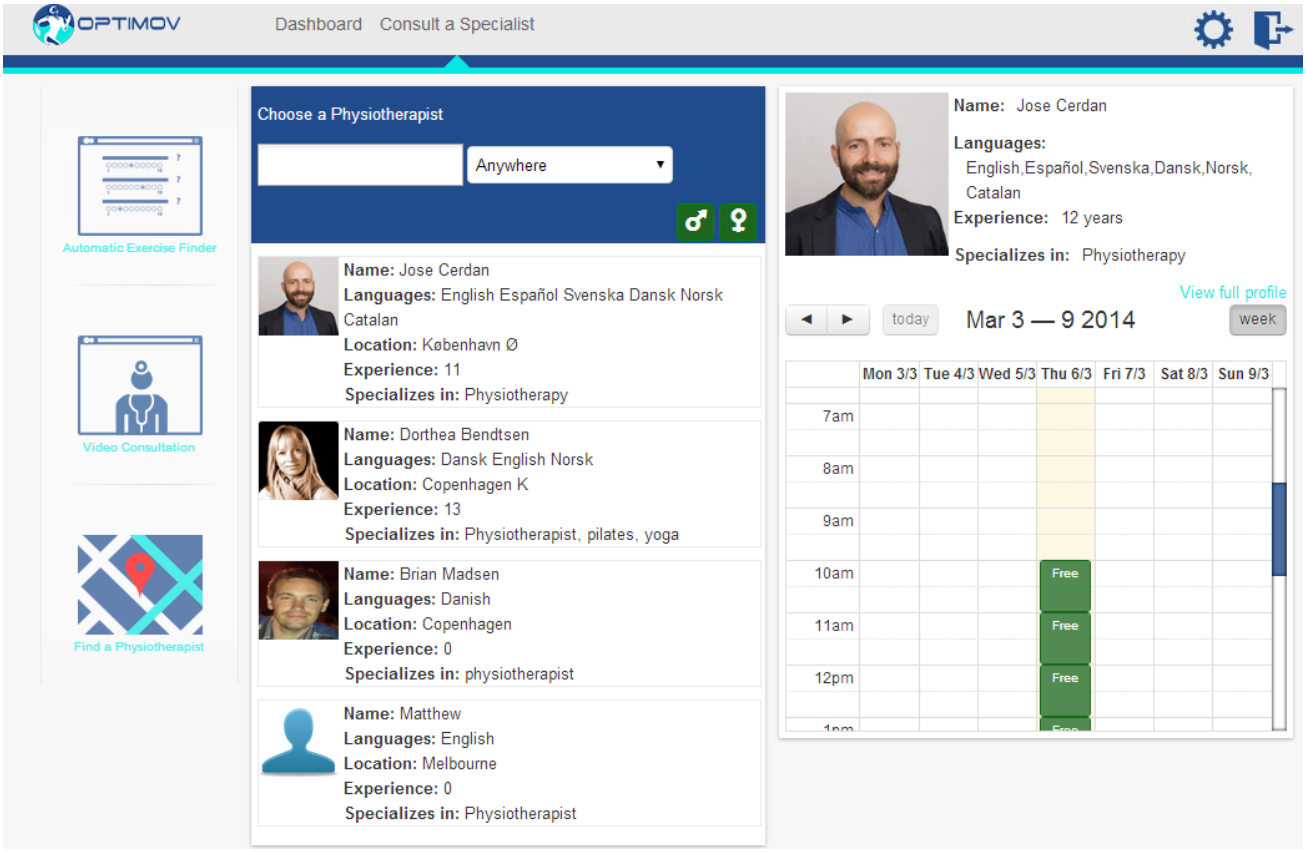
- Provides access to exercises anywhere and anytime so the patient can perform the exercises comfortably and when it fits his schedule.
- Offers audiovisual introductions to carry out the movements correctly beacause some exercise can be difficult to understand.
- Motivates the user to change his exercises or see his progress by providing an exercise schedule, tracking his progress, and keeping him in contact with the physiotherapist.
- Sustains motivation and makes rehabilitation fun with the 3D trainer, because these kinds of exercises can be quite boring without any videos.
- Provides the opportunity to continue to use the platform after the rehabilitation has been completed, looking at some graphics or keeping track of the next coming day of rehabilitation ensuring long-term progress.

b) A general overview

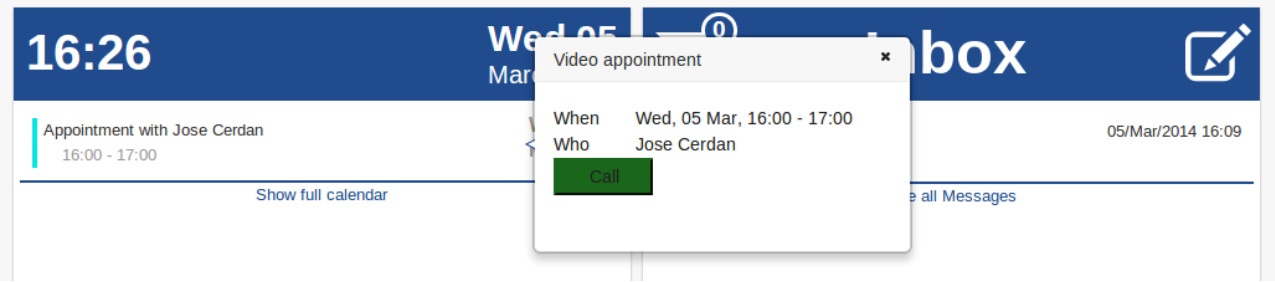
The schema below shows schematically the possibilities offered to a patient and the way the various functions relate for a typical user who searches for a therapist, schedules, an appointment, and who receives a course of exercises from a physiotherapist.



The user begins interacting with the system by selecting a specialist through the interface shown below; the specialist can be chosen from a list based on essential information (name, languages spoken, years of experience, specialization, etc.). Upon selection of a specialist, a frame in the right-hand side will show a calendar with his/her current appointments, from which the patient can schedule an appointment.



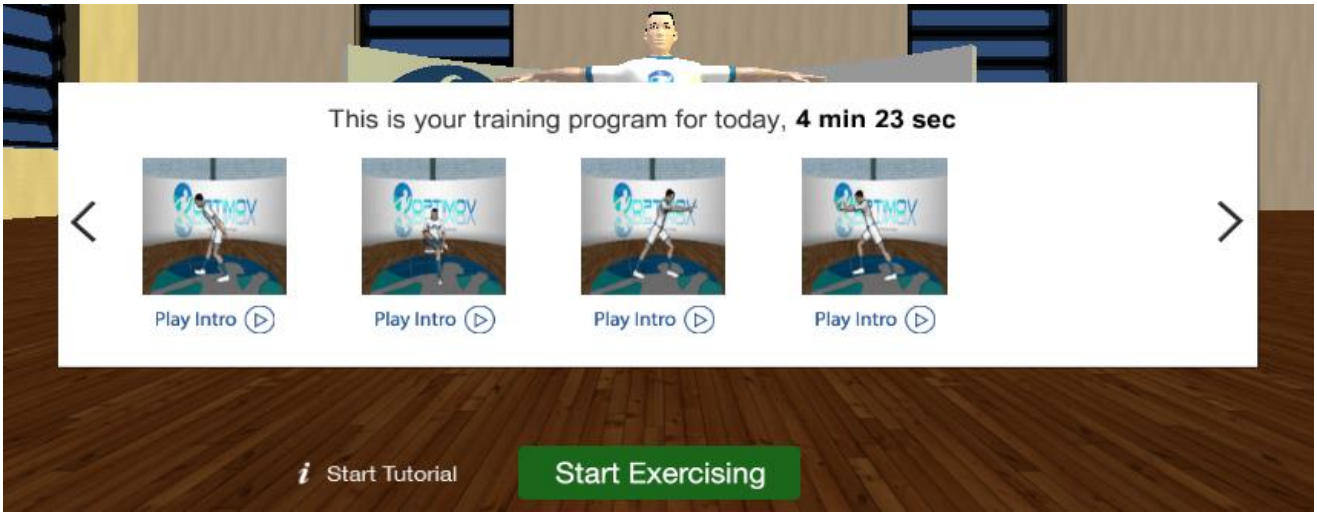
When the appointment is confirmed, it appears in the patient's calendar, and when that dates arrives, a pop-up appears and the patient has the option of calling his physiotherapist or just waiting for his call.



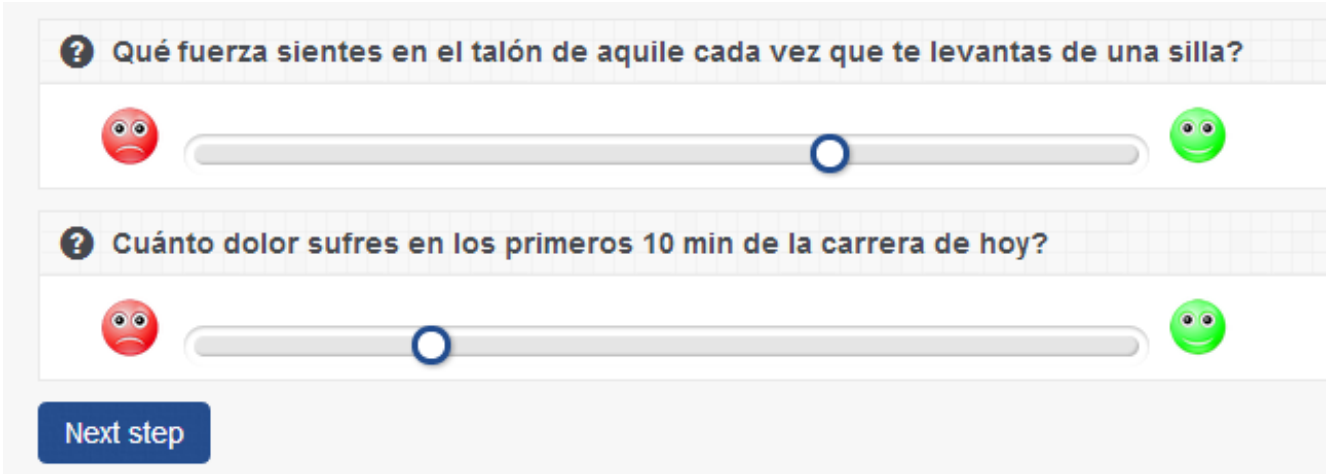
c) Dashboard

The Dashboard is the main page for the patient, where he can see his personal calendar, his inbox, and the progression or evolution of the pain. This graph some paragraphs down, depending on the answers to questions like answers to the physiotherapist's questions (see the graph in Figure 1 for an example of pain evolution). It is a great tool to determine whether the exercises are working on the patient. For example if the line would go down or would not change at all after a while, the physiotherapist would decide to change the set of exercises.

The following pictures show how the set exercises look. Each exercise can be watched in a small screen to show the user what he must do. Clicking in Start Exercising will open a big screen with the first exercises. Before to start the next exercise, there is a countdown or preparation which can be used as a break. Then, the next exercise will appear to have done all of them.



Upon terminating an exercise, the user answers the questions about how he feels after the exercises. These questions can also be shown before doing the exercises to give the physiotherapist more details about the pain.



The next graph is a portion of the interface with two graphs showing the evolution of pain for a patient. The yellow indicates the pain before doing the exercises, and the blue one indicates the pain just after the exercises. By analyzing this graph, the physiotherapist will know more about the pain, and have a direct and intuitive knowledge of whether and to what extent the exercises are working. This will allow the therapist to "fine tune" therapy based on the reactions of each specific patient.

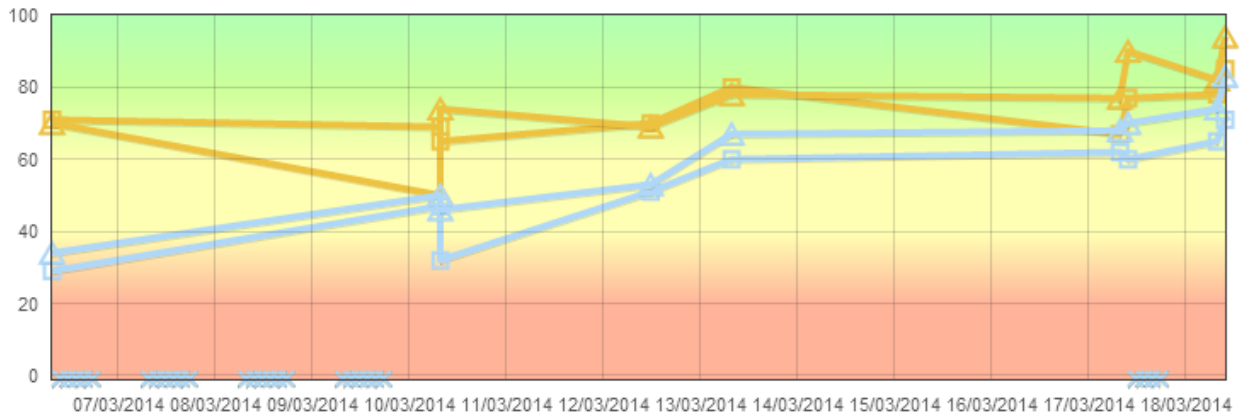


Fig. 1

3. A Brief History of blogging

The blog (a contraction of the expression 'binary log') is one of the most interesting and widespread means of expression of the pre-social-networks internet. Given the time of its first appearance (late 1990's to early 2000's) and its characteristics, it can be considered a link between the older form of understanding the internet – mainly a passive search for information – and the new one – a dynamic process in which people are at the same time producers and consumers of content.

Almost fifteen years from its First appearance, and amidst all the innovations brought by social networks, the success of the blog continues unabated, and blogs are still the main form of internet self-expression.

Although they had already been in common use for quite some years, the definite consecration of blogs in the public eye came in 2004, when both candidates to the presidency of the US made heavy use of them as a propaganda and fund-raising tool [1].

Businesses were quick to follow suit: they saw quite early the potential of blogs as a marketing tool. Rather than having a company describes the merit of a product (a strategy that has lost some of its effectiveness on buyers numbed by decades of advertising). They could rely on the opinion of the customers themselves – an opinion that, being given away for free and without the prospect of any financial gain, was often deemed more trustworthy than that of the companies.

Of course, this reliance on bloggers had also a somewhat perverse effect. Not only old we assist to the emergence of "star" bloggers (a phenomenon that, per se, violates the egalitarian nature of the medium), but companies have been known to offer monies to the most followed bloggers (VIP trip to their fashion shows, dinners, vacations…) with the intent (often, alas, successful) to bias their opinion [2]. A further limitation deriving from the explosive diffusion of blogs may be illustrated by paraphrasing an old saying: *"so many blogs, so little to say"*; finding something fresh and original to say on a topic is quite difficult, and the majority of blogs tends to concentrate on a few issues of popular cultures about which almost anything that can be said has

already been said. Consequently, the density of new information and new ideas in blogs is quite low [3].

These limitations notwithstanding, blogs are an attractive medium to express oneself in a semi-informal way. From the point of view of a business, one of the most appealing features of blogs is their "fidelity generating" effect: people are more likely to return and to spend on a web site on which they can post opinions that on a web site from which you simply get information. An open forum like a blog is also very useful from the point of view of the continuity of experience and responsiveness of the company: by listening to the opinions of its users, the company can detect and correct possible problems before they seriously disrupt the user's experience.

For all these reasons, Optimov saw the inclusion of a blog in his offering of services as a decision of strategic importance for the long-term success of its business.

The design and implementation of such a blog constitutes the bulk of this work.

4. Creating the Blog

a) Requirements

Optimov ApS did not provide many requirements for the blog because they wanted to give me responsibility to make some decisions in the company. The few requirements they had were pretty straightforward. During development, additional requirements were occasionally created based on the experience with early versions. They concerned mostly minor points and never touched upon the essential functionality of the blog, which remained pretty much the same during the whole development.

1) Types of users

Since the beginnings this point was pretty clear: there would be three kinds of users for the Blog, with different privileges: home-user, administrator, and super user.

- The home-user would be a normal user who would be allowed to read all entries and comments and write comments on a post.
- The administrators would be all the employees of the company. In addition to all the privileges of a normal user, the administrator can post a message but he has to wait to have the post approved by a super user.
- The super user would have the ability to approve or delete every single post and comment or everything at the same time.

2) Functional requirements

At the beginning, the company decided that in order to simplify the development and, especially, testing, there would be only one kind of user (what later became the super user). However, during development I decided that the small additional effort necessary to implement more kinds of users (all I had to do was to manage permissions according to user type) would pay off in terms of flexibility of the application, so I decided to implement the three kinds.

I decided to organize the development "back to front": first developed the back-end so that once I had it working, it would be easier to design and test the front-end. Finally, I would take the front-end of the main page and merge it with my code.

The back-end was developed mostly in Phython, with small portions written in PHP. The Database runs in phpMyAdmin or SQLite. I chose these applications as resources to implement the Database because my own laptop, where I had to work, it is quite limited to work in a high level. At the end, I was working with the SQLite but the last version was implemented with phpMyAdmin.

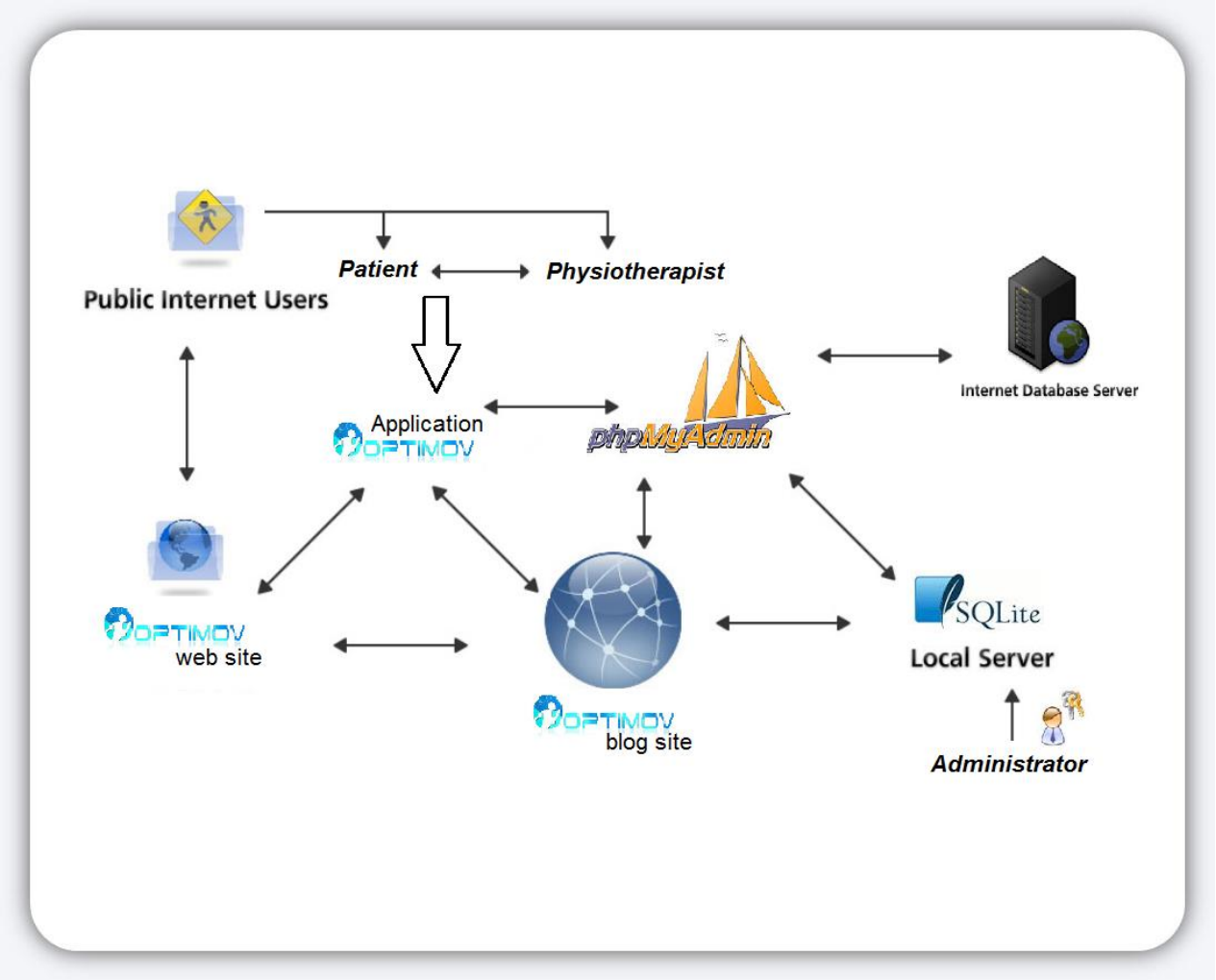
The front-end code would be in HTML, CSS and all languages that I would decide to add to get a much better final result.

b) Design

One of the requirements was that the interface includes a bag-cloud where we could see the most used tags of the blog, with the same color and different sizes.

Most of the design activity for this project was related to the back-end, as most of the design of the front-end was already done when I took over this project. During three days, Optimov's CTO, Tomasz Anielak and I designed together the Database of the blog which is explained later.

The following diagram illustrates the interaction between front-end and back-end, between the users and Optimov's application and blog; between these and the databases.



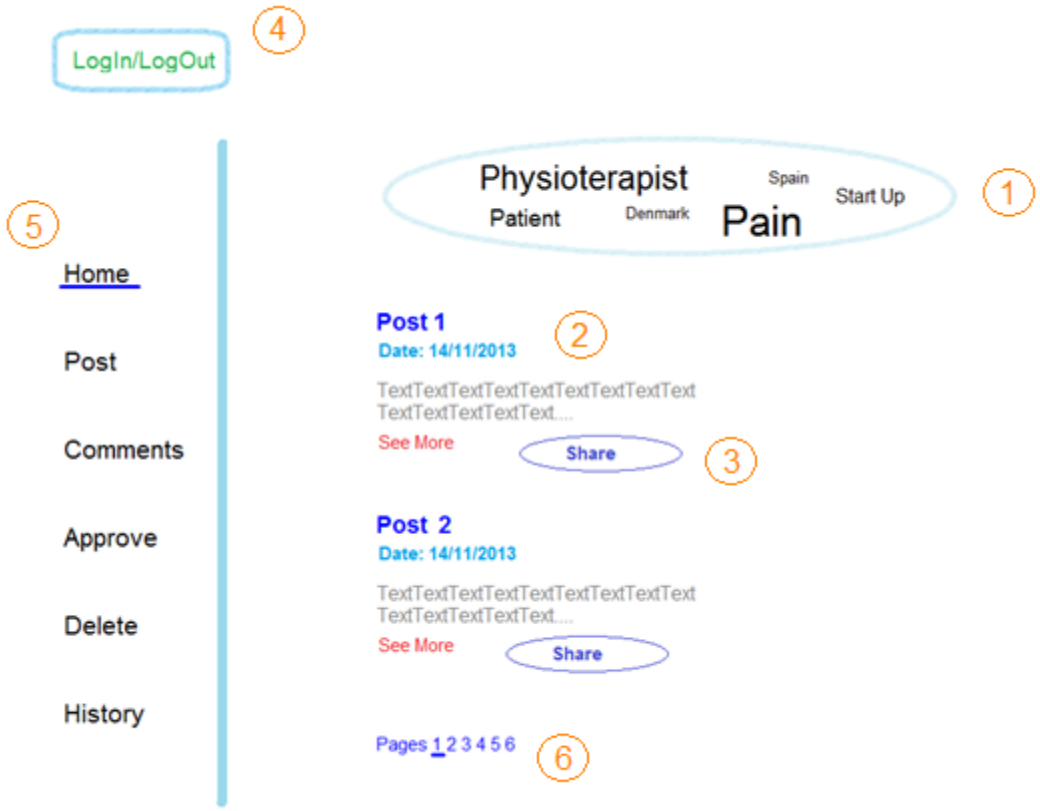
Then, the blog would be connected with everything on the Internet. For the first tests we used SQLite as local server, where only the administrator (in this case, myself) would have access. At the end of development, the blog would work with phpMyAdmin which gives us more services. Again, only the administrator or super user can have access to this application of the Database.

The home users –patients and physiotherapist—will have access to Optimov's blog as they have to Optimov's Application and its main page. If they want more privileges, e.q. to write a comment, they would have to register and login by their selves.

The blog, the application and the web site of Optimov will be connected each other, something that will be very helpful for the user.

1) Front-end

The layout for the front-end part for the blog is:



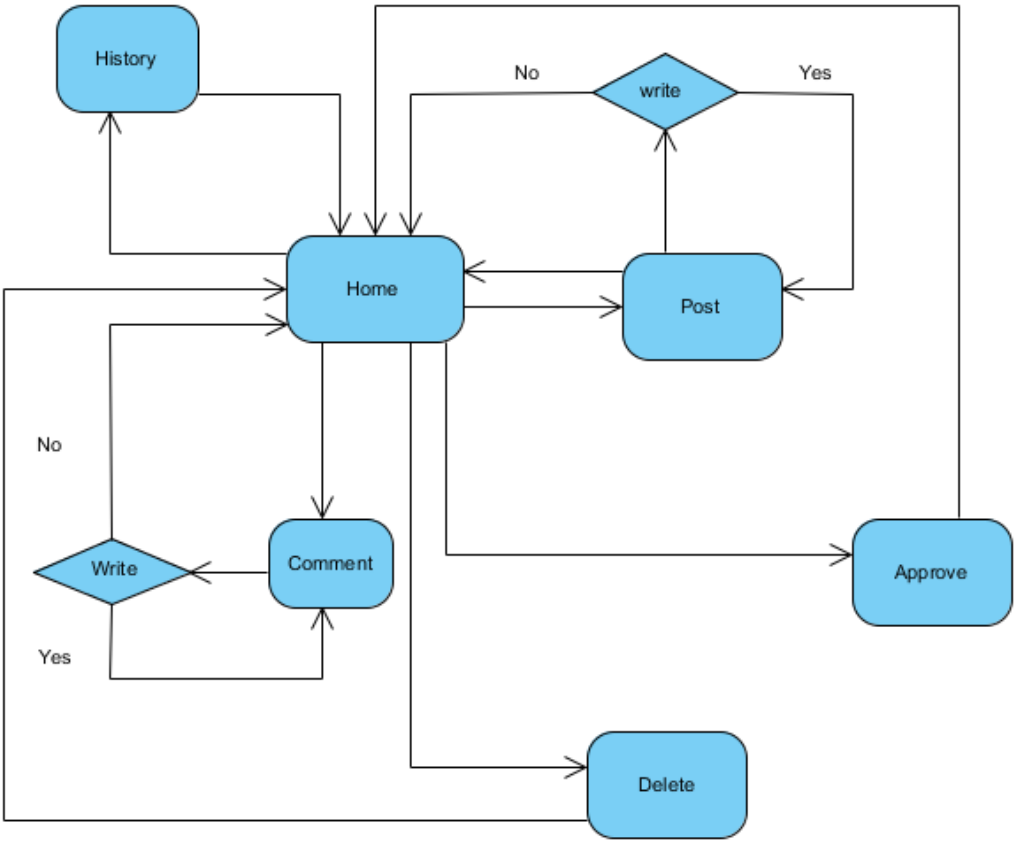
I made this design thinking that the Company needs an easy blog to use and followed by a clear interface.

Then, there would be a few changes on it, like the positions of each button or a little changes of its appearance. There is not more design pages because they decided that everything was clear enough and a front-end design is not useful at all.

The main elements are:

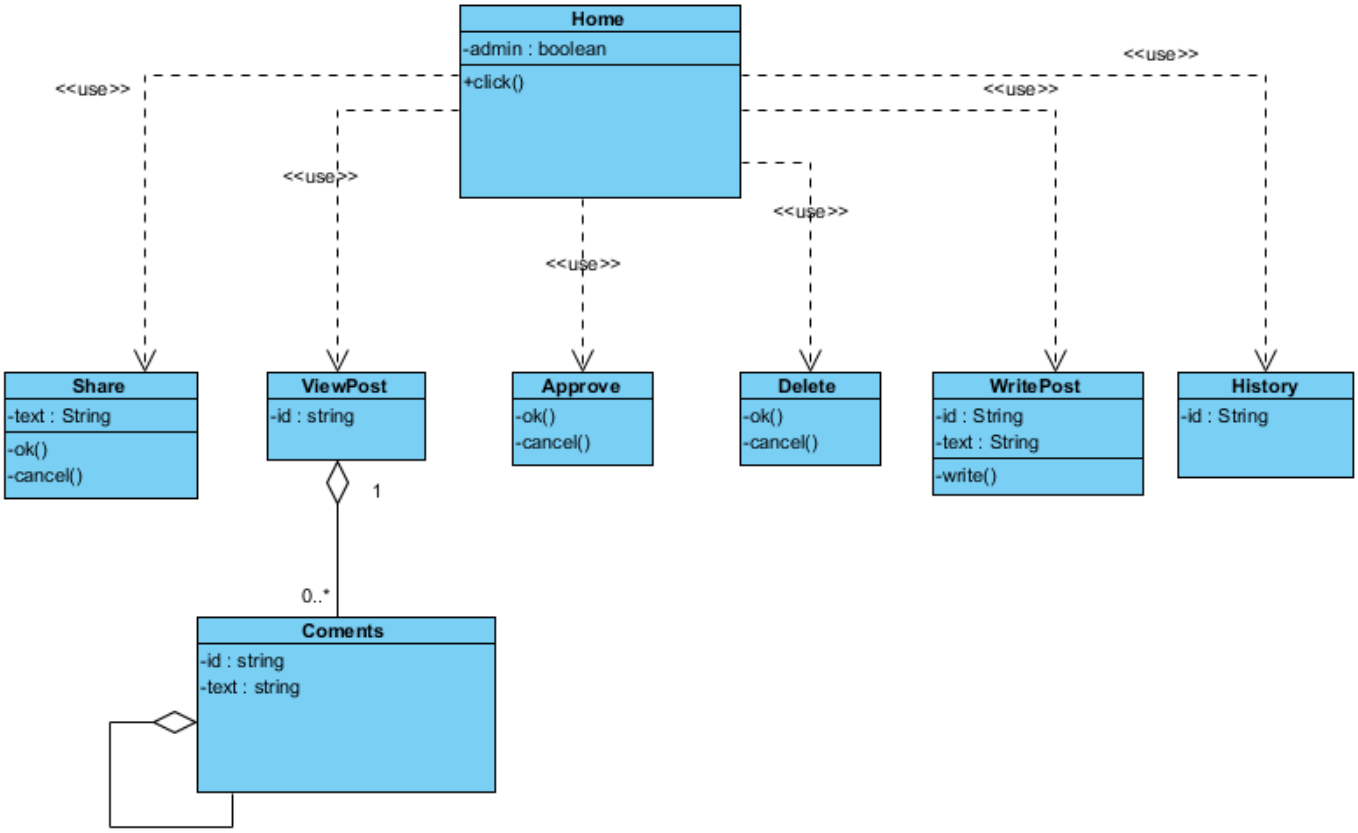
1) The cloud map. All tags are shown with different sizes and these tags are a link to all related posts.
2) The post with a few details, e.g. the title and the date. Clinking on ‘See More’, full text would appear.
3) Share buttons. It button was designed to share a post. Also it will be easy to add more social networks in the near future.
4) LogIn/LogOut. Where the user would start or close a session with this account.
5) Panel. To navigate around the blog
6) The number of all pages around all of them. In the design, each page would have two entries.

Activity diagram



Home is the main page, from this point we can go everywhere and come back. To write a comment or a post, we must confirm that we are sure about it, and then it is necessary to approve it or even delete it. The history will be all post and comments together.

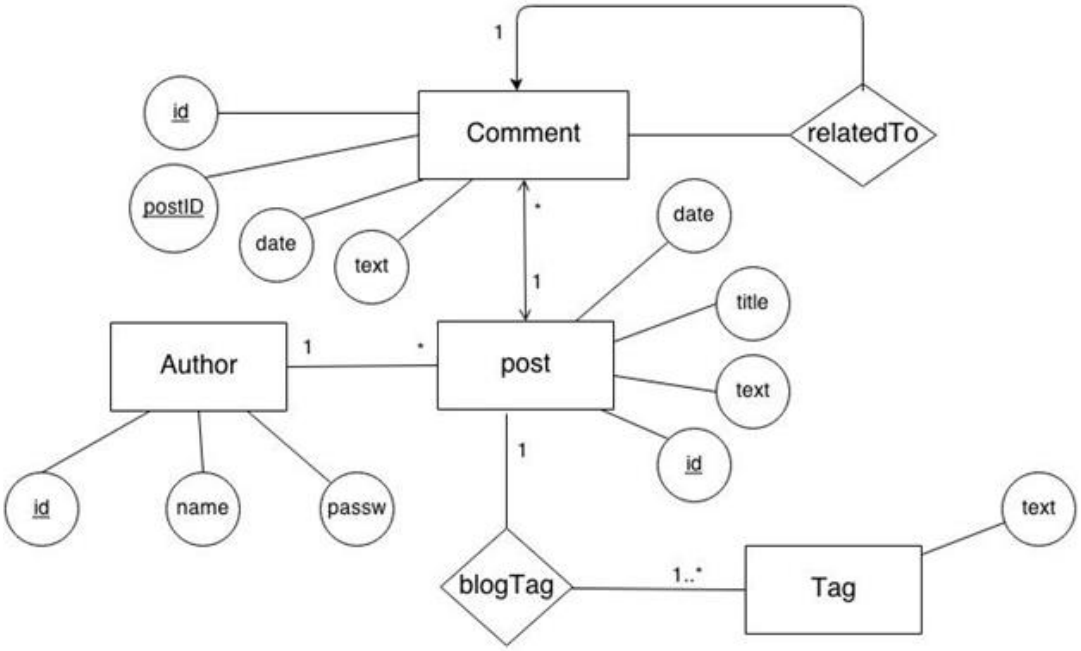
Class diagram



In the class diagram we can see that each function of the blog has a different class. All classes have different attributes and operations. For the comments we need an extra class because a comment can be a reply to another comment and there can be many comments in each post. It is important to note that there is an attribute --this attribute has been declarated as Boolean-- to know if the user is the administrator; in that case he will have more privileges, like access to approve or delete something.

2) *Functionality – back-end*

Data Base



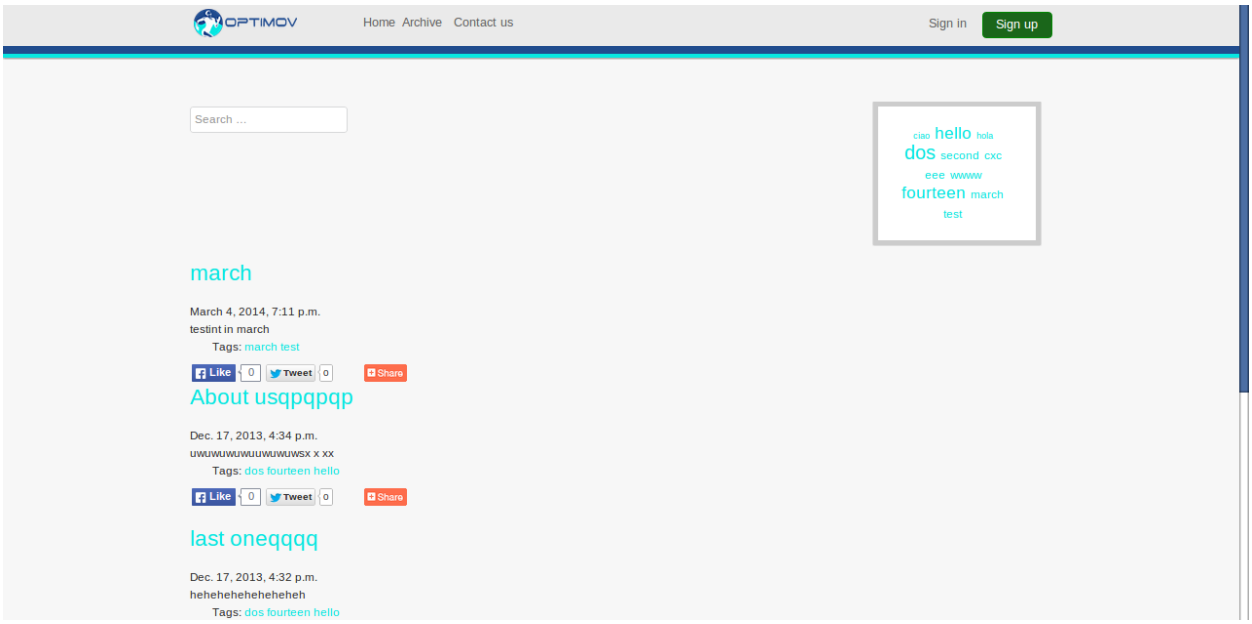
The entities of the diagram are Post, Comment, Author, Tag and Parent. Post is directly connected with Author and Tag. Every comment is connected with only one post. The key of Post is ID and there are two keys for Comment, which are id and postID. There is another different key called id, this key is for Author. Parent has an attribute called commentID which is the ID of a comment and each parent can have as many comments as are possible. Post is connected with Tag by the relation blogTag and each post should have at least a tag.

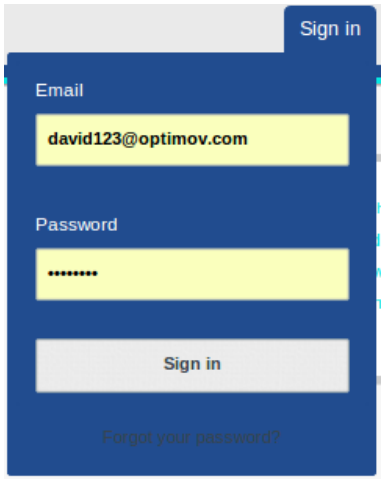
c) Implementation

In this section, we will see how the final version of the blog looks and its different properties and points are completed. I will explain also how the cloud map and other important functions and algorithms were implemented.

1) Front-end

The first view that the user gets is the beginning of the blog without any access, as we can see in the next screen shot. If the user does not log in, he can have access only to Optimov's web site, the main page of the blog, the search, the archive or history of all posts, and our 'Contact Us'.



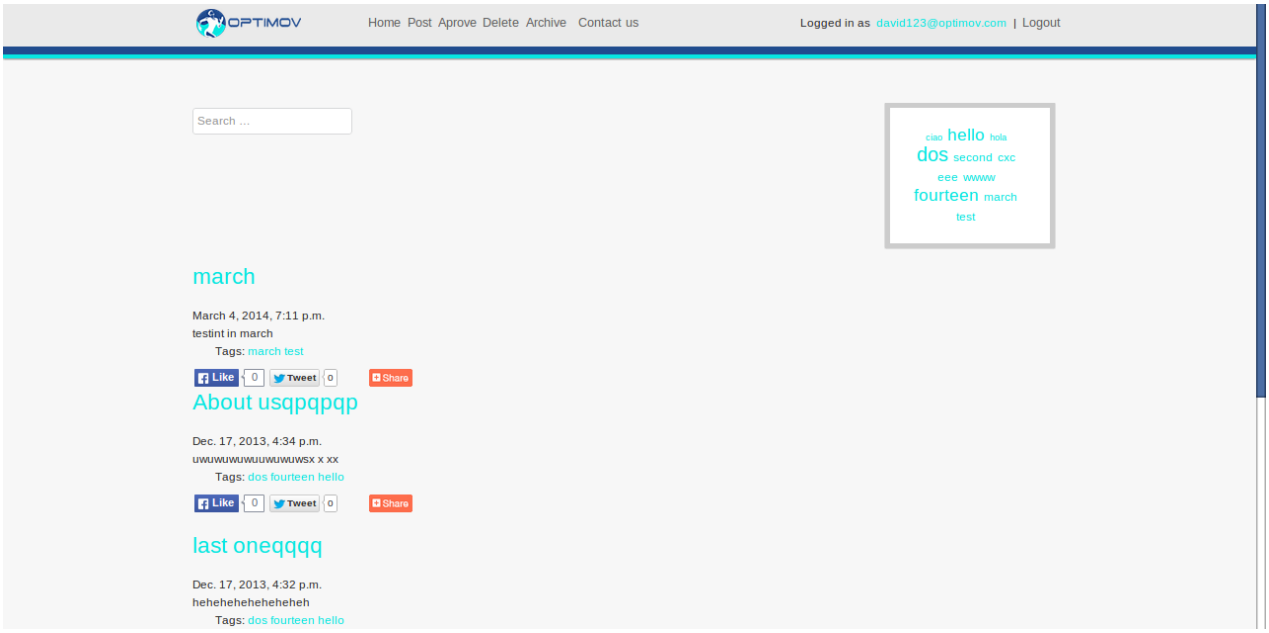


If we sign in as super user, we automatically have full access to the functionality of the blog.

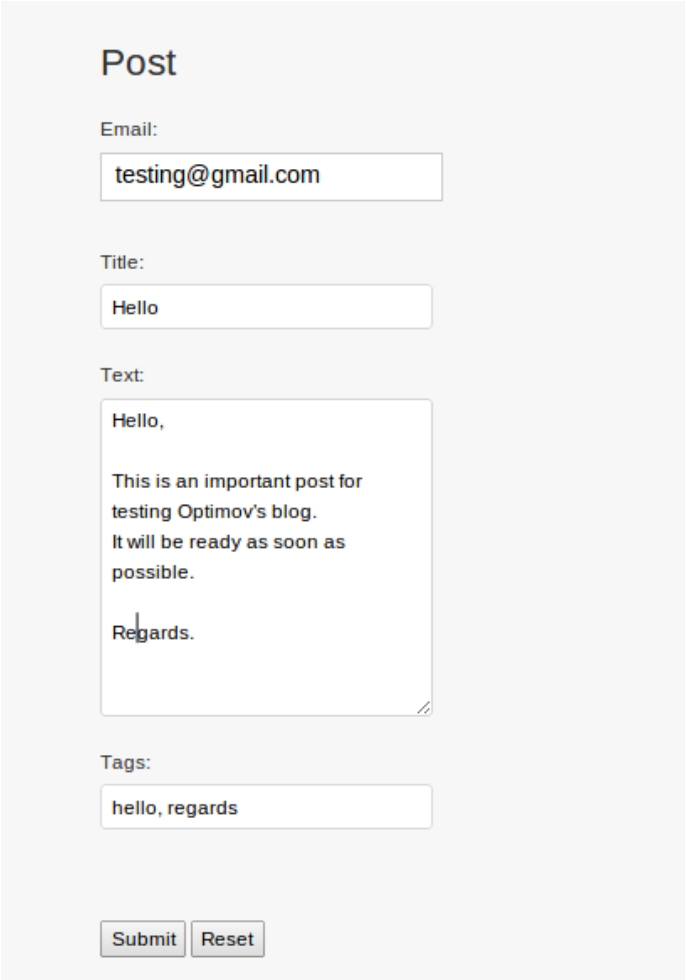
We decided to design the blog with the same appearance as the main page. This way, it will be more familiar to the user, and much easier to develop because it was done by taking all CSS files that Optimov has saved and then I just had to merge everything.

After this merger, we had to fix some small problems like the position of the blue lines since they were switched, the size of the logo and the zoom of the scrollbar, etc.

In the page shown below there are a maximum of four entries, and there is a navigation button to go to the next or the previous page.



i) Post



To post something it is necessary to have an e-mail, a title, some text and tags. It is important to have in mind that the tags must be separated by commas.

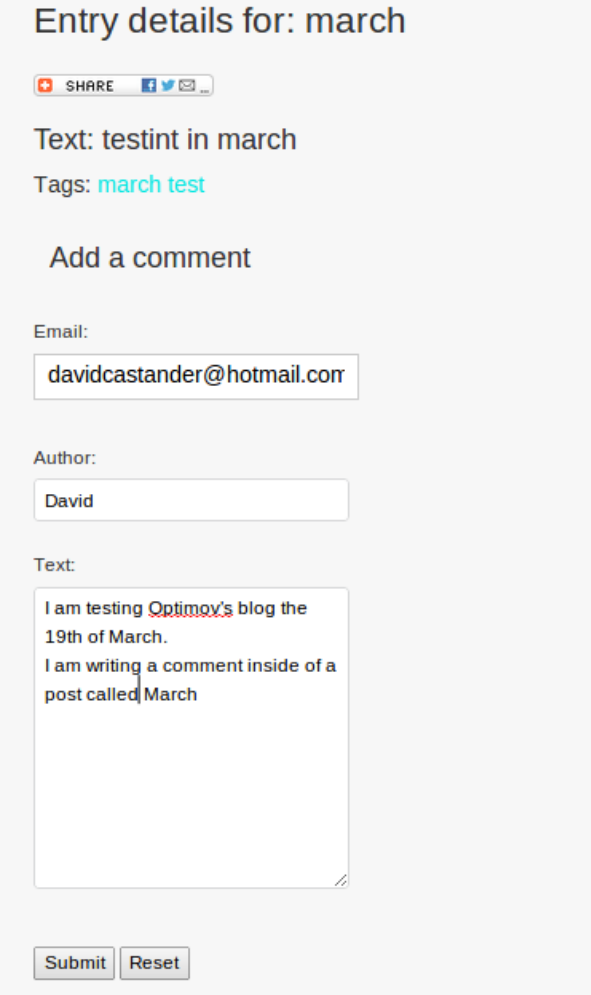
The company has decided not to have any control over the email and will probably be deleted in the future because the user has already signed in, so we would know who posted it. At the moment, the author of the post is never shown, and whoever posts something on this blog will be someone who is working for Optimov ApS. Therefore, it is the responsibility of the user to post something with the right email.

I return all tags and the title associated by the code followed below. Then all these data will be saved into a variable called ‘t’.

```
return render_to_response('success_posted.html', {'tag':tags, 'title':title},
context_instance=RequestContext(request))
```

The first argument is where we would go later; the second one is data that we need to send; to talk about the third argument, we should first know that Django comes with a special Context class, *django.template.RequestContext*, it takes an *HttpRequest* as its first argument and it also automatically populates the context with a few variables, according to our *TEMPLATE_CONTEXT_PROCESSORS* setting. All arguments passed to the function are attributes of the data base.

ii) Post details and comment



Clicking on the title of each post, allows us to see the details of that post.

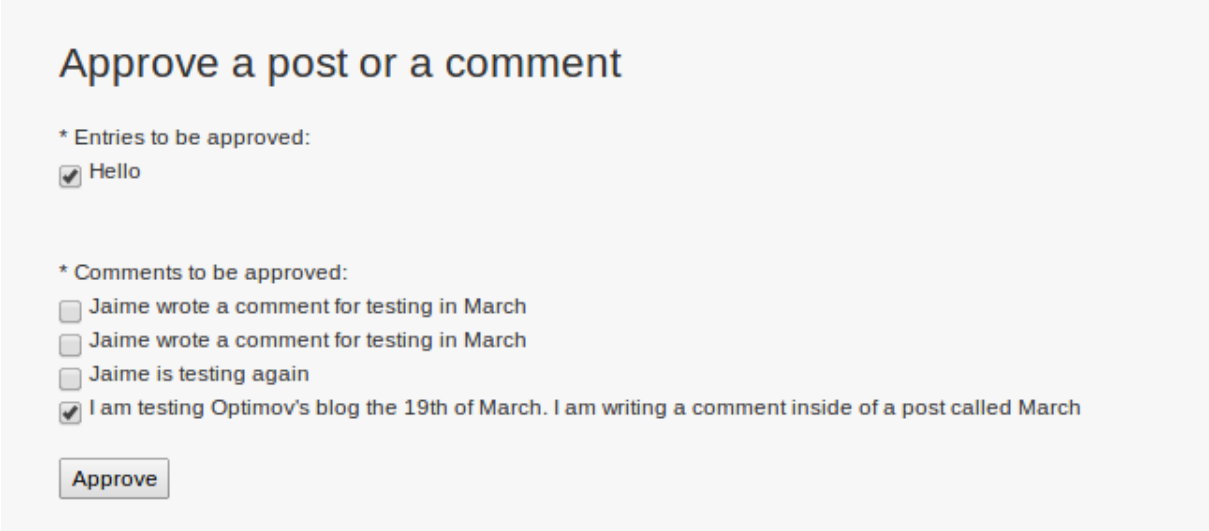
As we see in this picture, we can leave a comment. The comments will not have any tag, just the email, its author and the text.

Then someone could reply to our comment, and we could start a conversation on this comment.

iii) Approve

To post or comment on something, it is necessary that the administrator of the blog or super user approves it, even if it is himself who wants to post something.

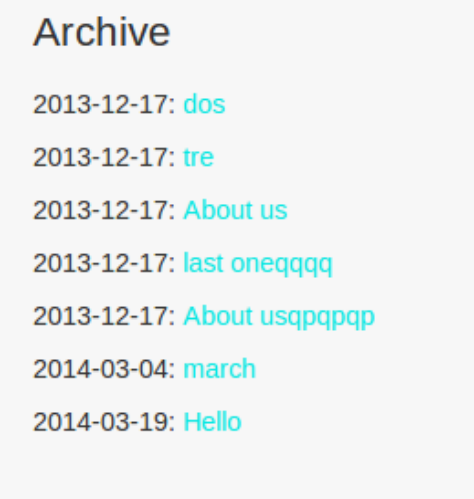
He can approve as many entries or comments as he wants.



iv) Delete

The page to delete the posts or comments has exactly the same appearance as the approval one

v) Archive



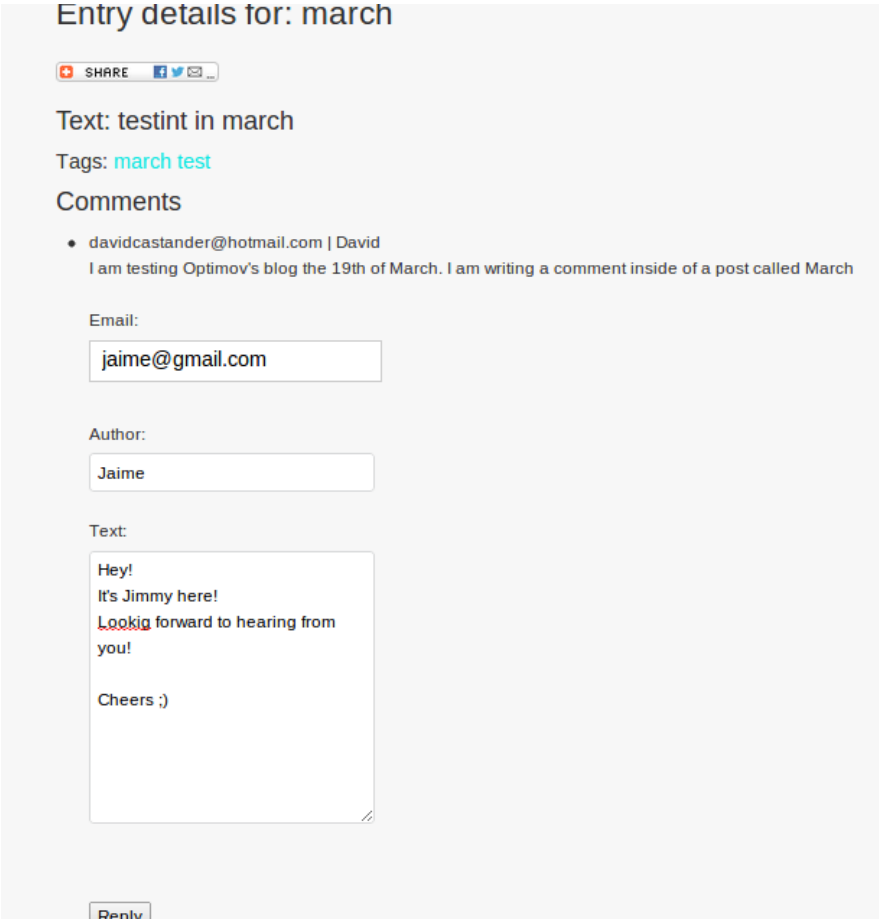
This section of the blog shows the history of all posts that have been approved. The user can look for a post and go there quickly.

We only show the date and the title of each post. Clicking on the title brings us again to the details of that post. Optimov wants to have a simple archive, and that is why we only see the title and date.

vi) Post proof

Here we have the proof that the post created before it was approved and posted correctly.

vii) Reply

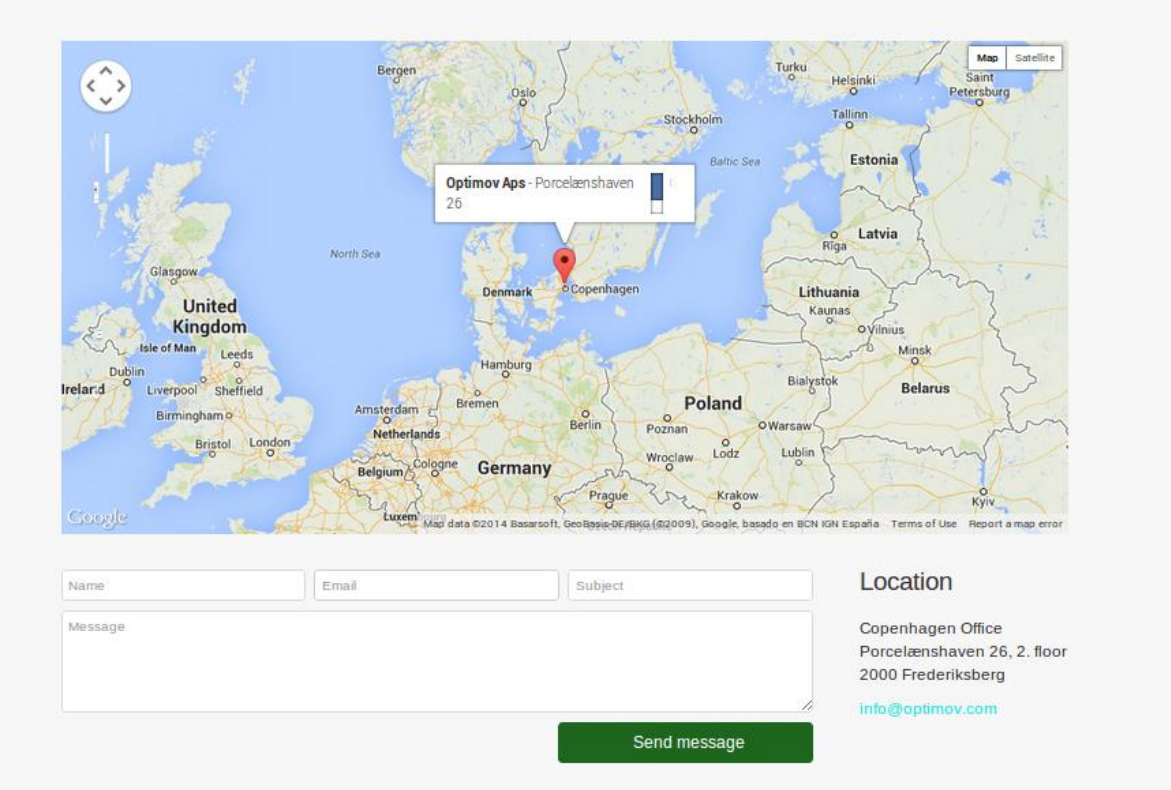


Within the reply to a comment, we will write a comment in response to another comment.

Then it will look like the typical conversation, which is easy to follow.

viii) Contact Us

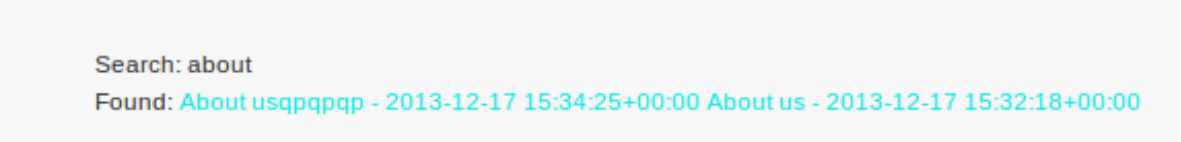
At first, we got a screen where the user could read about all members working at the Company, an email and all social networks to contact Optimov ApS. Then we changed it to have the option of sending an email directly to info@optimov.com, but this option turned out to be not as useful as we thought, even when the functionality worked. Therefore, for the last version of the blog, we decided that the implementation of this section should be exactly the same as the view we have in the Optimov's web and which is showed below. In conclusion, the implementation of 'Contact Us' was clear and easy, because I only had to copy and paste the code already implemented and supervised by Tomasz Anielak, Optimov's CTO, who coded this.



ix) Search

In the current version, the user can search based on title but not on tags. If we want to look for a tag, we should first change the whole algorithm of search and remake the whole functionality of saving the tags since the function called Taggit does not work in the case that we want. In the future we will look for an alternative to Taggit and solve this problem.

The view that we get after a search in'About us' of all the posts with that title and their dates. Clicking on each one opens its details.

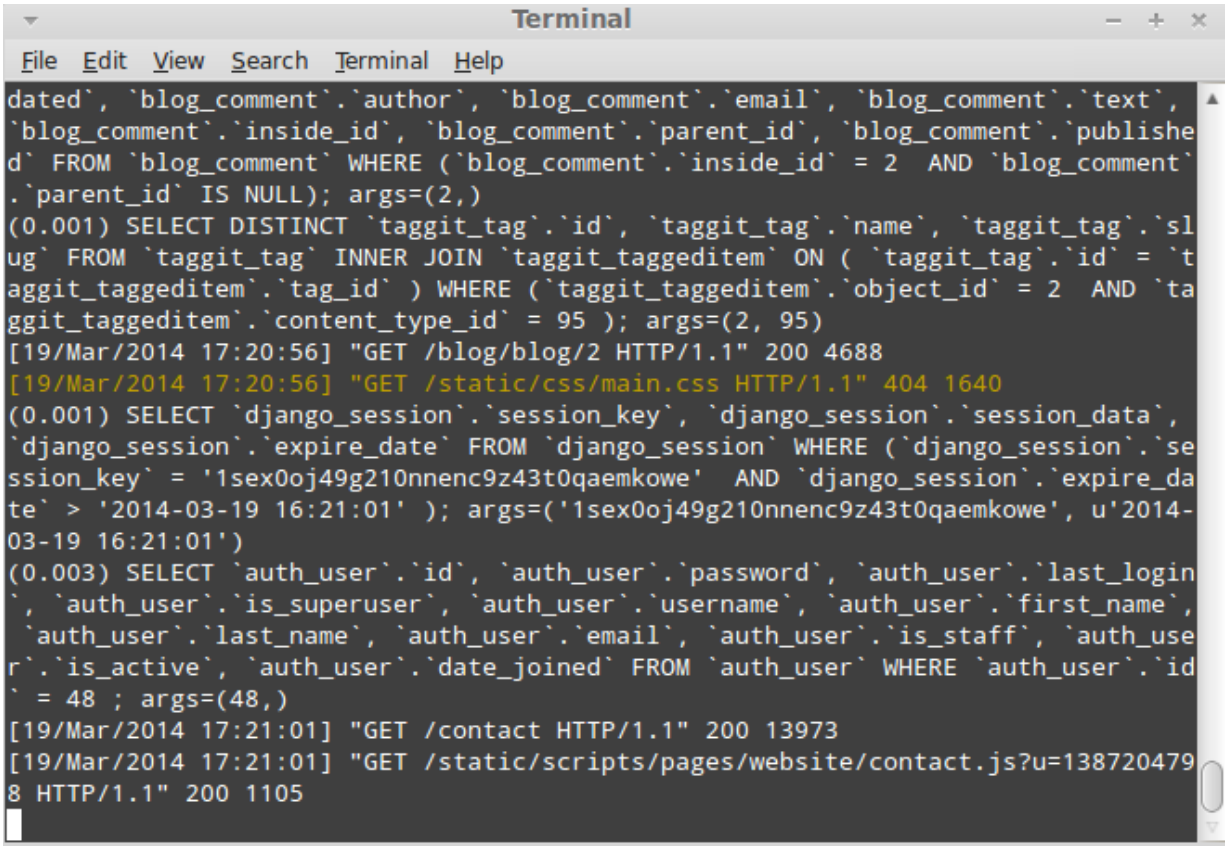


About the code of this function is in a file called 'views.py' file and in the function 'search'. The main code is the following:

```
if ('s' in request.GET) and request.GET['s'].strip():
    query_string = request.GET['s'] #text got it
    entry_query = get_query(query_string, ['title', 'text',])
found_entries = Entry.objects.filter(entry_query).order_by('-id')
```

Firstly, we check that the user wishes to search for something, then we got that text that the user has written and we go into the query taking that string written. Outside of the loop, we show what we get ordered by the IDs.

x) Signals



Each time the user does something or navigates around the blog, the server catches different signals. This extra feature of the blog was done with the tests and the future in mind. For example, as we can see, the password, or other arguments, has been encrypted previously to send that signal, and each signal has a different command associated with it.

xi) Cloud

The cloud is a map of all tags saved in all posts with different sizes. The more tags we save with the same name, the more the size of the tag grows.

We thought that we should put the cloud on the top of the web page just below the search, but after taking a look at other blogs, we decided that we should locate the cloud on the top right of the blog like most of the others.

- ***Cloud Algorithm***

The code of this algorithm is located into the file 'views.py' and then into the function 'index' due to we show the cloud map in the main page of the blog.

The most important part of the algorithm is clear and simple as we see in the next textbox

```
tagc = Tag.objects.all()

tags = []

for tag in tagc:

        count = TaggedItem.objects.filter(tag=tag).count()*3 + 10

        tags.append((tag,count))
```

The size of the count was calculated by trial and error to get the most fixed size for Optimov.

I save the tags inside of ‘ctx’, this variable is an array which contains the page, the entries and other stuff that we return by the next line of code:

```
render_to_response('blogIndex.html', ctx, context_instance=RequestContext(request))
```

Then, inside of the file models.py and the function ‘Entry’, I initiated the tags with the function TaggableManager(), otherwise Iwould not be able to work with the tags like I do in ‘views.py’.

2) Functionality – back-end

Data Base

The pictures below show the databases of entries, comments and Taggit or cloud map.

i) Entries

Each entry has an ID as a primary key, created and updated dates, a title, the text saved, which is much longer than the title, and whether the post has been already published. The title and text could be empty if the user decides it.

	id	created	updated	title	text	published
Edit Inline Edit Copy Delete	2	2013-12-17 15:31:15	2013-12-17 15:31:15	dos	Hello, here you are the 2nd post	1
Edit Inline Edit Copy Delete	3	2013-12-17 15:31:48	2013-12-17 15:31:48	tre	qqqqqqq lslslslsls	1
Edit Inline Edit Copy Delete	4	2013-12-17 15:32:18	2013-12-17 15:32:18	About us	qoqoqooqoqoqoqoqoq	1
Edit Inline Edit Copy Delete	5	2013-12-17 15:32:49	2013-12-17 15:32:49	last oneqqqq	hehehehehehehehehe	1
Edit Inline Edit Copy Delete	6	2013-12-17 15:34:25	2013-12-17 15:34:25	About usqpqpqp	uwuwuwuwuuwuwuwsx x xx	1
Edit Inline Edit Copy Delete	7	2014-03-04 18:11:45	2014-03-04 18:11:45	march	testint in march	1

ii) Comments

Like the entries, each comment has an ID as primary key, create and update dates, a title and the text saved, plus its author and the appropriate email. The field *inside_id* signifies the entry with which it is associated and *parent_id* means the comment with which it is associated. These fields are foreign keys, since they are probably IDs for another table. The user can reply to each comment and see them in a tree structure.

←T→	id	created	updated	author	email	text	inside_id	parent_id	published
Edit Inline Edit Copy Delete	3	2013-12-20 18:30:39	2013-12-20 18:30:39	david	davidcastander@hotmail.com	Testing testing	6	*NULL*	1
Edit Inline Edit Copy Delete	4	2014-03-05 13:26:33	2014-03-05 13:26:33	Jaime	jaime@gmail.com	Jaime wrote a comment for testing in March	7	*NULL*	0
Edit Inline Edit Copy Delete	5	2014-03-05 13:27:05	2014-03-05 13:27:05	Jaime	jaime@gmail.com	Jaime wrote a comment for testing in March	7	*NULL*	0
Edit Inline Edit Copy Delete	6	2014-03-05 13:27:27	2014-03-05 13:27:27	Jaime	jaime@gmail.com	Jaime is testing again	7	*NULL*	0

iii) Cloud map

The database of the cloud map has only an ID as primary key, a name and at least a tag; in the database we have called the tags as slug and the test as name, just to make it different with the test, in our case, the test and the tag will be probably always exactly the same.

As we can see in the next picture, the ID is just the order of how we have saved the tags.

	id	name	slug
Edit Inline Edit Copy Delete	1	ciao	ciao
Edit Inline Edit Copy Delete	2	hello	hello
Edit Inline Edit Copy Delete	3	hola	hola
Edit Inline Edit Copy Delete	4	dos	dos
Edit Inline Edit Copy Delete	5	second	second
Edit Inline Edit Copy Delete	6	cxc	cxc
Edit Inline Edit Copy Delete	7	eee	eee
Edit Inline Edit Copy Delete	8	wwww	wwww
Edit Inline Edit Copy Delete	9	fourteen	fourteen
Edit Inline Edit Copy Delete	10	march	march
Edit Inline Edit Copy Delete	11	test	test

5. Tests

a) Users

The blog was tested for ten users in total, all of them with a different role in the company and in the blog to best simulate all possible conditions that the blog will be used for.

b) Platform

The blog was tested first in a local host and then in the main server of Optimov ApS. Then we had to delete it from the server, so now it only works in local.

6. Extra development

a) Activity prior to this project

The first weeks at Optimov, I started giving development support to improve the web site and its application. I received small tasks, and after two weeks there were no tasks left for me but we still wanted to work together. Optimov's CEO, Johannes Scheibe told me to "try to find something to do on your own—something like a small project—then you could easily use your work at Optimov for your thesis in Spain." In that moment I remembered that having a blog is becoming more and more popular in all companies around the world. I told Optimov my project idea, they liked it, and we decided to make the blog.

Some of the tasks I was working on are:

- About Us. Photos of all members and a funny interaction to see the names and their positions.
- Save/Finish buttons. When you click on one of these buttons, the information only has to be saved one time and not as many times as the user clicks the button.
- Button to install Plugin. When Unity plugin is uninstalled, a button or a message appears in the middle of the screen.
    a) For the first time a new exercise is used, it is necessary to refresh.
    b) To check if the plugin is ready.

If the plugin is not ready, to show the message and the button to refresh the page

- There were so many music files. I had to find a solution to use less space for music and keep as much quality as possible. It was necessary to change the properties of the files using different programs like mp3-quality-modifier. At the beginning, the music files were 40Mb size, and at the end we used 12MB of music.
- Buttons: I am a Therapist, I am a Patient
- Sign-Up needs to be made more obvious.
- Sign Up button had to be much bigger
- There should be just a checkbox : "I am a Therapist"
- Delete sign in and login as tags.

- Login is a link and Join the Optimovement is a button.
- A visual confirmation that sign-up was successful

b) The blog: Present

Optimov's blog is completely ready to be used. It is not online yet because they are focusing on the application, payments and other important tasks.

On other hand, they are also discussing what they will post in their blog and when it is the best moment to do it. Since they do not yet have enough followers on Twitter or on Facebook.

Optimov ApS has decided to wait to launch the blog until they launch the full version of the application. They have to focus on solving the different problems with the application because those problems are the priority and the blog is not a priority yet.

c) The blog: Future

As soon as the full version is launched, Optimov ApS will grow a lot. That moment will be perfect to launch the blog. The blog will become very useful not only for Optimov's customers, but also for everybody interested in physiotherapy in any way. Optimov will post not only news about themselves and their application, but also news related to physiotherapy, alternatives, exercises, etc.

The issues that we will have to develop are first, the option to do a search for a tag and second, a way to show the comments, because the structure of each comment should appear similar to a tree.

7. Conclusions and Future work

The final vesion of the blog works very well, it is stable and it satisfies all requirements or tasks. I hope that it will be used as a tool for Optimov ApS to get more users in their services and to give more information to everybody about injuries, treatments and other news around the field of physiotherapist.

It is not necessary to add many more functionalities into the blog, one of them might be to do more efficient the algorithm for searching something, specially to search tags. There could be a system integrated to calculate points, for example, as many comments, shares or likes as the user would have, more chances to get prizes like sessions for free with a physiotherapist with Optimov's application or the chance to write a post on Optimov's blog.

Other alternative for the future of the blog it could be to develop a special version for mobile phones, the text of the post would be much shorter.

By doing this project, I have learnt new programming languages and new work methods. I have also learnt a lot within myself and from all team members. I think I have improved my social and programming skills. I have lived a great experience abroad and I now I have a better English and more friends and work connections around the world.

Anexo

Quick install guide – Django

Before you can use Django, you'll need to get it installed. We have a complete installation guide that covers all the possibilities; this guide will guide you to a simple, minimal installation that'll work while you walk through the introduction.

Install Python

Being a Python Web framework, Django requires Python. It works with Python 2.7, 3.2, 3.3, or 3.4. All these versions of Python include a lightweight database called SQLite so you won't need to set up a database just yet.

Get the latest version of Python at http://www.python.org/download/ or with your operating system's package manager.

Django on Jython

If you use Jython (a Python implementation for the Java platform), you'll need to follow a few additional steps. See Running Django on Jython for details.

You can verify that Python is installed by typing python from your shell; you should see something like:

```
Python 3.3.3 (default, Nov 26 2013, 13:33:18)

[GCC 4.8.2] on linux

Type "help", "copyright", "credits" or "license" for more
information.

>>>
```

Set up a database

This step is only necessary if you'd like to work with a "large" database engine like PostgreSQL, MySQL, or Oracle. To install such a database, consult the database installation information.

Remove any old versions of Django

If you are upgrading your installation of Django from a previous version, you will need to uninstall the old Django version before installing the new version.

Install Django

You've got three easy options to install Django:

- Install a version of Django provided by your operating system distribution. This is the quickest option for those who have operating systems that distribute Django.
- Install an official release. This is the best approach for users who want a stable version number and aren't concerned about running a slightly older version of Django.
- Install the latest development version. This is best for users who want the latest-and-greatest features and aren't afraid of running brand-new code.

Always refer to the documentation that corresponds to the version of Django you're using!

If you do either of the first two steps, keep an eye out for parts of the documentation marked **new in development version**. That phrase flags features that are only available in development versions of Django, and they likely won't work with an official release.

Verifying

To verify that Django can be seen by Python, type python from your shell. Then at the Python prompt, try to import Django:

```
>>> import django
>>> print(django.get_version())
1.8
```

You may have another version of Django installed.

That's it!

That's it – you can now move onto the tutorial.

References

[1] L. A. Adamic and N. Glance. The political blogosphere and the 2004 U.S election: divided they blog. In LinkKDD ’05: Proceedings of the 3rd international workshop on Link discovery, pages 36–43, 2005.

[2] Rachel Dodes “Bloggers get under the tent” The Wall street Journal, Sep. 12, 2006

[3] Flora S. Tsai, Agus T. Kwee. (September, 2011). Database optimization for novelty mining of business blogs. Expert Systems with Applications: An International Journal. 38 (Issue 9), 11040-11047.

[4] http://www.djangobook.com/en/2.0/index.html

[5] https://docs.djangoproject.com/

[6] http://www.sqlabs.net/sqlitemanager.php

[7] http://www.youtube.com/watch?v=BwVv91xIMnI&list=PL385A53B00B8B158E

[8] https://www.youtube.com/watch?v=yAyWM-iEIos&list=PL313A26329E0C8900

[9] https://www.youtube.com/watch?v=D5VlpgEVVg4

[10] https://www.youtube.com/watch?v=cQ_hUFdYiDM

Contraportada

 /optimov

 /Optimov